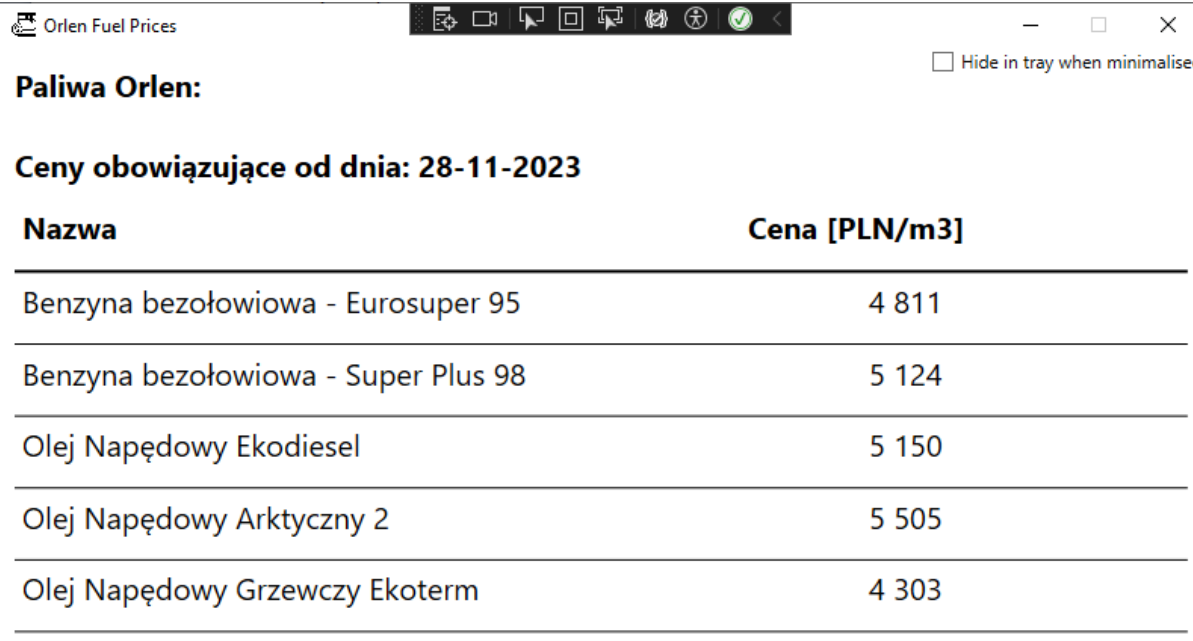


# Aplikacja Orlen Fuel Prices

## Dokumentacja

Kacper Mąkosa 4P1



Paliwa Orlen:	
Ceny obowiązujące od dnia: 28-11-2023	
Nazwa	Cena [PLN/m3]
Benzyna bezołowiowa - Eurosuper 95	4 811
Benzyna bezołowiowa - Super Plus 98	5 124
Olej Napędowy Ekodiesel	5 150
Olej Napędowy Arktyczny 2	5 505
Olej Napędowy Grzewczy Ekoterm	4 303

Ogólny wygląd aplikacji

### Działanie aplikacji:

Aplikacja WPF wykorzystująca metodę "Web Scraping" wykorzystując silnik Selenium oraz język C#, aby pobrać dane ze strony internetowej:

[https://www.ortalen.pl/pl/dla-biznesu/hurtowe-ceny-paliw?fbclid=IwAR0JYK4fmnVGIG6MtKgd9AKW\\_ziZVEe7hJOUZaXYA2R3gVFZOCVQjmurk48](https://www.ortalen.pl/pl/dla-biznesu/hurtowe-ceny-paliw?fbclid=IwAR0JYK4fmnVGIG6MtKgd9AKW_ziZVEe7hJOUZaXYA2R3gVFZOCVQjmurk48)

Aplikacja wyświetla się w zasobniku systemowym, można również zaznaczyć opcję aby aplikacja schowała się do zasobnika po kliknięciu na przycisk minimalizuj.

Ceny obowiązujące od dnia: 28-11-2023	
Nazwa	Cena [PLN/m3]
Benzyna bezołowiowa - Eurosuper 95	4 811
Benzyna bezołowiowa - Super Plus 98	5 124
Olej Napędowy Ekodiesel	5 150
Olej Napędowy Arktyczny 2	5 505
Olej Napędowy Grzewczy Ekoterm	4 303

Wygląd aplikacji wyświetlanej w zminimalizowanej wersji po naciśnięciu ikonki w zasobniku systemowym.

## Klasa DataScraper.cs

### Opis klasy:

Klasa zawierająca główną funkcję dotyczącą “Web scrapingu”, tj. pobierania danych ze strony orlenu.

### Funkcje:

```
public void Scrapedata(string url, ChromeOptions options)
```

Funkcja pobiera dane ze strony orlenu, wyświetla je w aplikacji przekazuje do zmiennej scrapedData, która jest wykorzystywana aby wyświetlać dane w “balonie”, w funkcji również wywoływana jest funkcja SaveScrapedData, która zapisuje dane do

tabeli w bazie danych.

```
Odwolania: 2
public void Scrapedata(string url, ChromeOptions options)
{
    using (var driver = new ChromeDriver(driverService, options))
    {
        driver.Navigate().GoToUrl(url);
        try
        {
            Thread.Sleep(5000);
            IWebElement dateElement = driver.FindElement(By.XPath("//p[@class='hcp__content-date']/STRONG"));
            Debug.WriteLine("Connection Status: Connection went successfully");
            if (dateElement != null)
            {
                string content = dateElement.Text;
                date = content;
                mainWindow.Dispatcher.Invoke(() =>
                {
                    mainWindow.DateText.Visibility = Visibility.Collapsed;
                    mainWindow.RowsStackPanel.Children.Clear();
                    var textBlock = new TextBlock
                    {
                        Text = "Paliwa Orlen:\n\n" + "Ceny obowiązujące od dnia: " + content,
                        Margin = new Thickness(0, 0, 0, 10),
                        FontFamily = new FontFamily("Segoe UI"),
                        FontSize = 20,
                        FontWeight = FontWeights.Bold,
                        Foreground = Brushes.Black
                    };

                    mainWindow.RowsStackPanel.Children.Add(textBlock);
                });

                scrapedData.Add("Ceny obowiązujące od dnia: " + content);
            }
            else
            {
                System.Windows.MessageBox.Show("Date element not found");
            }
        }
    }
}
```

Zrzut ekranu części funkcji.

## Klasa DataBaseManager.cs.

### Opis klasy:

W klasie zadeklarowana jest funkcja SaveScrapedData, wykorzystująca atrybuty, string name, int price, string date, które podawane są przy wywoływaniu tej klasy. Prosta funkcja zapisująca pobrane dane do bazy danych jeśli są one nowe.

```
public void SaveScrapedData(string name, int price, string date)
{
    try
    {
        using (var connection = new SQLiteConnection(connectionString))
        {
            connection.Open();
            using (var countCommand = new SQLiteCommand("SELECT COUNT(*) FROM ScrapedData WHERE date = @date", connection))
            {
                countCommand.Parameters.AddWithValue("@date", date);
                int rowCount = Convert.ToInt32(countCommand.ExecuteScalar());

                if (rowCount < 5)
                {
                    using (var insertCommand = new SQLiteCommand("INSERT INTO ScrapedData (name, price, date) VALUES (@name, @price, @date)", connection))
                    {
                        insertCommand.Parameters.AddWithValue("@name", name);
                        insertCommand.Parameters.AddWithValue("@price", price);
                        insertCommand.Parameters.AddWithValue("@date", date);
                        insertCommand.ExecuteNonQuery();
                    }
                }
                else
                {
                    Console.WriteLine("Cannot insert more than 5 records with the same date.");
                }
            }
        }
    }
}
```

## Klasa CustomBalloon.cs

### Opis klasy:

Klasa zawierająca funkcje tworzącą "Balon", który wyświetla się po naciśnięciu na ikonkę w zasobniku systemowym.

### Funkcje:

private void InitializeComponent(),

Funkcja inicjalizująca balona.

```
1 odwołanie
private void InitializeComponent()
{
    grid = new Grid();
    grid.RowDefinitions.Add(new RowDefinition { Height = GridLength.Auto });
    var balloonBorder = new Border
    {
        Background = Brushes.White,
        CornerRadius = new CornerRadius(5),
        BorderBrush = Brushes.Black,
        BorderThickness = new Thickness(1),
        Child = grid
    };
    Content = balloonBorder;
}
```

private TextBlock CreateMessageTextBlock(string message)

Funkcja tworząca blok tekstowy dla balona.

```
1 odwołanie
private TextBlock CreateMessageTextBlock(string message)
{
    var messageTextBlock = new TextBlock
    {
        FontSize = 16,
        TextWrapping = TextWrapping.Wrap,
        Margin = new Thickness(10),
        HorizontalAlignment = HorizontalAlignment.Left,
        Foreground = Brushes.Black,
        Text = message
    };

    return messageTextBlock;
}
```

## Klasa MainWindow.xaml.cs

### Opis klasy:

Klasa zawiera inicjalizację wszystkich funkcji zawartych w innych klasach, oraz kilka dodatkowych funkcji.

### Funkcje:

#### private void SetupNotifyIcon()

Funkcja ustawiająca wyświetlanie w trayu.

```
1 odwołanie
private void SetupNotifyIcon()
{
    notifyIcon = new TaskbarIcon();
    notifyIcon.Visibility = Visibility.Visible;
    string iconPath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "crude-oil.ico");
    notifyIcon.Icon = new System.Drawing.Icon(iconPath);
    notifyIcon.ToolTipText = "Orlen Fuel Prices\n" + "Left click to show prices. " + "Double left click to open app";
    notifyIcon.TrayMouseDoubleClick += NotifyIcon_DoubleClick;
    notifyIcon.TrayLeftMouseUp += NotifyIcon_LeftClick;
}
```

#### private void TenMinuteTimer()

Funkcja ustawiająca timer na 10 minut i pobierająca dane za każdym razem po upływie czasu.

```
1 odwołanie
private void TenMinuteTimer()
{
    timer = new DispatcherTimer();
    timer.Interval = TimeSpan.FromMinutes(10);
    timer.Tick += (sender, e) =>
    {
        RowsStackPanel.Children.Clear();
        DateText.Visibility = Visibility.Visible;
        DateText.Text = "Aktualizacja w toku..";
        ScrapedataAsync();
        timer.Start();
    };
    timer.Start();
}
```

#### private void Scrapedata()

Funkcja wywołująca funkcję Scrapedata z klasy DataScrapper.cs z odpowiednimi atrybutami.

```
1 odwołanie
private void Scrapedata()
{
    string url = "https://www.ornen.pl/pl/dla-biznesu/hurtowe-ceny-palim?fbclid=IwAR0JYK4fmmVGLG6MtKgd9AKW_ziZVEe7hJOUZaXYA2R3gVFZOCVQjmurk48";
    ChromeOptions options = new ChromeOptions();
    options.AddArguments("--enable-gpu", "--enable-background-tracing", "--window-size=1920,1080", "f'user-agent={agent}'", "headless");
    options.PageLoadStrategy = PageLoadStrategy.Normal;
    dataScrapper.Scrapedata(url, options);
}
```